IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	Veil et al.			
Serial No.	:	09/855,898	Examiner	:	Fowlkes, Andre R.
Filed	:	May 15, 2001	Group Art Unit:		2192
For	:	METHOD AND SYSTEM FOR CONDITIONAL INSTALLATION AND EXECUTION OF SERVICES IN A SECURE COMPUTING ENVIRONMENT			

DECLARATION PURSUANT TO 37 C.F.R. § 1.131

I hereby certify that this paper is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

November 4, 2005
Date of Deposit

Eric J. Faragi
Attorney Name

51,259
PTO Registration No.

Signature

November 4, 2005
Date of Signature

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

We, Leonard Scott Veil and Erica Elisabeth Tups, hereby declare as follows:

1. Leonard Scott Veil is a joint inventor of the inventions as recited in claims 1-43 of the above-identified patent application.

2. The inventions of claims 1-43 of the above-identified patent application, among others, were conceived and reduced to practice in this country, prior to March 19, 2001 (the "Critical Date"), which is the filing date of U.S. Patent Application Pub. No. 2004/0015961

entitled "Method and Apparatus for Automatic Prerequisite Verification and Installation of Software" of Thomas E. Chefalas (the "Chefalas Reference").

3. Leonard Scott Veil had the inventions of claims 1-43, among others, reduced to practice prior to the Critical Date by implementing the concepts of the method and system recited in these using an EMBASSY system, and successfully testing them thereafter also prior to the Critical Date.

4. The announcement of the demonstration of the EMBASSY system, *i.e.* the Trusted Client PC solution, at the Fall 2000 COMDEX show is attached herewith as Exhibit A. The techniques specified in claims 1-43 of the above-identified application, among others, were embodied in the demonstrated EMBASSY system.

5. During the 2000 Fall COMDEX Convention, Wave Systems demonstrated its EMBASSY system. This demonstration took place as planned, November 13 – 17, 2000, at the 2000 COMDEX in Las Vegas, Nevada. Wave's demonstrations took place at booth L1573, North Hall, Las Vegas Convention Center. This demonstration of Wave's EMBASSY system included the EMBASSY Network Server (ENS) software, EMBASSY hardware devices and the EMBASSY Operating System. The demonstration proved the existence of the systems, methods and apparatus of claims 1-43 of the above-identified application, among others, and their ability to work for their intended purpose(s).

6. The demonstration utilized the system hardware and software of both the server and client components of the EMBASSY system showing the successful reliance of the demonstration on the ability of the system to perform all of the functionality described in claims 1-43 of the above-identified application, among others. The demonstration could not have successfully operated if the implementation of these claims were not reduced to practice and

reproducible. This system implemented and utilized the concepts of secure conditional loading of applets based on the availability of device resources. Specifically, applets requiring the availability of secure input (keypad) and secure output (LCD display) resources on the EMBASSY device were securely downloaded to the device and, only after the device and its software (operating system) verified the capabilities of the device with respect to the resource requirements contained within the securely downloaded meta-data appended to the applet, would the EMBASSY device securely install the applet and execute it; as it did in the demonstration.

7. In order to successfully demonstrate the system, adequate testing was performed in preparation for the demonstration. The demonstration and the testing associated with it represented actual conditions. That is, the system was at a sufficient level of development that it operated per the claims 1-43 of the above-identified application, among others. Extensive testing of the system was performed prior to the demonstration as part of the development process used to create the system. Additional testing was performed in order to assure a successful demonstration at COMDEX.

8. The test results clearly demonstrated that prior testing of the system was performed successfully, thus providing a successfully running system for the COMDEX demonstration. These demonstrations were performed numerous times each day during the COMDEX convention.

9. Attached herewith as Exhibit B is a PowerPoint presentation prepared by Len Veil, dated September 1, 2000, which is an overview of the EMBASSY system. The feature of secure, conditional loading of applets is contained therein.

10. Attached herewith as Exhibit C is an internal Wave document, the Invention Disclosure Form (IDF), dated October 18, 2000. The IDF provides a summary and explanation of certain features of the invention.

11. Attached herewith as Exhibit D is the announcement of the EMBASSY Application Developer's Kit, dated August 22, 2000, which implements certain features of the invention.

12. Attached herewith as Exhibit E are portions of the EMBASSY Aplication Developer's Guide, dated February 2001, showing the description of conditional resource loading.

13. Attached herewith as Exhibit F is a Wave Internal Engineering Status Report, dated November 2000, providing status on the work efforts preparing the system (development and testing) for COMDEX.

14. Attached herewith as Exhibit G are various system specifications predating the critical date.

15. I declare further that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful statements may jeopardize the validity of the application or any patent issuing therefrom.

Date: 11/4/05

Leonard Scott Veil

LEONARD SCOTT VEIL

wave

AMD and Wave Systems Demonstrate Trusted Client Computer Solutions For Secure E-Commerce and Services at Fall COMDEX

Las Vegas – November 13, 2000 Wave Systems Corp. (NASDAQ:WAVX) and AMD (NYSE:AMD) today announced they will demonstrate their first implementation of the Trusted Client PC solution, at the Fall 2000 COMDEX show.

Wave and AMD, based on a joint development agreement signed earlier this year, are developing a Trusted Client reference platform that is planned to enable greater security to be delivered to the PC.

The AMD and Wave reference platform is designed to be compliant with an evolving specification for PC trust and security developed by the Trusted Computing Platform Alliance (TCPA). Furthermore, AMD and Wave's solution adds significant additional functionality including user privacy options, distributed e-commerce and transaction capability. The open, programmable security hardware infrastructure has been designed to provide PC OEMS a template for building cost optimized Trusted Client PCs.

Secure applications to be demonstrated on the AMD/Wave Trusted Client platform at COMDEX – all powered by Wave's EMBASSY technology – are Secure Boot and TCPA integrity metrics , the Cyber-COMM secure e-commerce solution, and a metering application to support various commerce models for consumer entertainment content.

"Clearly the next wave of e-commerce will require personal computers and other platforms to protect the privacy and security of the consumer, including the protection of credit card numbers and other sensitive information, as well as protect digital content and enable such new services as pay-per-view and full function free trials," said Richard Heye, vice president and general manager, Texas Microprocessor Division (TMD).

The EMBASSY (EMBedded Application Security SYstem) Trusted Client platform is Wave's implementation of Trust @ the Edge, a new architecture for the Internet that moves core trust and security functions out to the edge of the Internet, and into end-user devices such as PCs, and peripherals. EMBASSY is an open and programmable hardware security co-processor subsystem designed to provide a platform capable of hosting secure application processing, and access to secure resources such as storage, time, cryptographic and key management. EMBASSY is platform independent, with robust interfaces and can be shared by multiple parties for trusted applications.

"By creating the framework to deliver security, trust and a broad range of new e-commerce service functions to PCs and other platforms, AMD and Wave are demonstrating leadership in transforming computing and the Internet," said Steven Sprague, president and CEO, Wave Systems. "With the EMBASSY Trusted Client architecture we are making it possible to provide users with the privacy options and security they want in their PC, protect digital goods and services for providers and enable new models of e-commerce."

The extended set of capabilities enabled by the Trusted Client architecture are designed to deliver a wide range of new content and services to users. By using any high speed network and Wave's Trusted Client architecture, consumers may be enabled to take advantage of a number of new distribution and buying models for content including rent-to-own, pay-per-view, as well as free trials of full-function software, games, music and videos. Businesses may eventually benefit from enhanced security in the use of emerging net-based tools such as Application Service Provider software delivery.

The solution developed by Wave and AMD will be demonstrated at Fall COMDEX, Las Vegas, November 13-17 in Booth L1573, North Hall, Las Vegas Convention Center.

About Wave Systems

Wave Systems' goal is to build a worldwide network of users based on trusted electronic relationships. Trust @ the Edge defines a new architectural model for the Internet, which embeds trust and security in every user device. Wave Systems is developing, deploying and licensing its EMBASSY Trusted Client technology for the mass adoption of this revolutionary model, integrating industry standard functions, from a wide range of partners that enable reliable, secure digital exchange and commerce over the Internet. At its core, Wave Systems is building the services, and enabling 3rd parties to build services that will take advantage of this new, open, Trust @ the Edge model.

For more information about Trust @ the Edge and Wave Systems visit: www.wave.com .

About AMD

AMD is a global supplier of integrated circuits for the personal and networkedcomputer and communications markets with manufacturing facilities in the United States,

Europe, Japan, and Asia. AMD produces microprocessors, flash memory devices, and support circuitry for communications and networking applications. Founded in 1969 and based in Sunnyvale, California, AMD had revenues of $2.9 billion in 1999. (NYSE: AMD).

For more information about today's announcement, please visit our virtual pressroom at http://www.amd.com/news/virtualpress/index.html. Additional press releases and information about AMD and its products are available at: http://www.amd.com/news/news.html.

Safe Harbor for Forward-Looking Statements Except for the statements of historical fact, the information presented herein constitutes forward-looking statements within the meaning of the Private Securities Litigation Reform Act of 1995. Such forward-looking statements involve known and unknown risks, uncertainties and other factors which may cause the actual results, performance or achievements of the company to be materially different from any future results, performance or achievements expressed or implied by such forward-looking statements. Such factors include general economic and business conditions, the ability to fund operations, the loss of market share, changes in consumer buying habits and other factors over which Wave Systems Corp. has little or no control.

Wave Corporate Contact:
Wave Systems Corp.
John Callahan
413-243-7029
Email: jcallahan@wavesys.com

Wave PR:
Brodeur Worldwide
Alex Thorne, 202-756-1600
Email: tmcmahon@brodeur.com

Wave Investor Relations:
Jaffoni & Collins
David Collins, 212-835-8500
Email: wavx@jcir.com

Close this window

Stay up-to-date on news relating to Wave. Join our email list by putting your email address in the space below.

join

Embassy

Embassy Partner Presentation
09.01.00
Wave Systems Corp

Note: this presentation contains material
that is proprietary and confidential to Wave Systems.
Disclosure outside of Wave Systems
personnel requires the execution of an NDA

Wave®

Applets

- Embassy applets are developed using an Embassy Applet SDK
- Each applet is compiled for execution on the Embassy target hardware device (native code)
- Future applet support may provide for hardware independent execution (JAVA or other OOP)
- Applet resource requirements are explicitly stated in the toolkit and instantiated in the applet code
- Applets are authorized to be used in the Embassy environment by a Certifying Agent
- Applet developers are responsible for import/export by appropriate government agencies

Wave®

Embassy Applet Construction

- Header contains necessary bookkeeping information
  - Version
  - Applet ID
  - Resource Requirements
  - Number of pages
  - Security classification
  - Command table
- Certifying Agent attest to applet validity by signing applet (header & plaintext executable)
  - Creates a certificate for every applet
- Signature is used to validate applet authenticity

Applet Context Table

	Hash of Code	Hash of Data	Flags	

- What is it?
  - It is a structure that binds a verified and permissioned applet executable and its associated persistent state into the run-time environment of a specific Embassy Device.
- What does it contain?
  - Applet ID
  - Hash of the Applet Code and Applet Data
  - Resources required to execute the Applet
  - Swap Key is a randomly generated key to encrypt the applet code and data
  - Flags for identifying a revoked applet or an applet that is invalid to execute.
- When it is produced
  - At applet installation, by the Embassy Device

Wave®

Applet Requirements

- Applet resource requirements are explicitly stated by the Certifying Agent and contained within the applet. Applet resource checking is performed during installation/loading. If the device does not contain the appropriate available resources, the applet is not installed/loaded
- A command handler within an applet cannot span multiple applet-pages
- Command handlers cannot invoke other applets
- Applet and applet-pages are host controlled
- An applet can lock a resource and then be swapped out. Only the applet or the Embassy OS may unlock the resource
- Resources cannot be shared amongst applets
- Commands sent from the host must be acknowledged by a command handler within the applet. Non-responding applets will be unloaded by the Embassy OS
- An "install-applet" is responsible for installing all applets in the system, and this "install-applet" will verify the applet by digital certificates and signatures before the installation process can complete

Security Model - Requirements

- All inputs to the Embassy device are verified by the Embassy OS for proper construction
- All inputs to the applets are verified by the applets for format and trustworthiness
- All trusted information stored outside the security boundary is appropriately encrypted and signed
- Applets cannot access data of other applets unless explicitly given permission to do so
- Applets cannot access code of other applets
- Applets do not assume anything about the Embassy device environment between invocations
- Applet resource requirements are known to the OS
- Applets cannot be installed unless the resource requirements are met
- Applets cannot be executed unless the required resources are available
- The Embassy OS prevents denial of service attacks by applets which monopolize resources
- The Embassy OS is responsible for clearing applet resources upon applet termination
- Public Key cryptography is used to protect Embassy Server, Embassy Device, and Certifying Agent keys
- Public Key cryptography is used to protect Applet Installation
- The Secure Time cannot be modified by applets

WAVE SYSTEMS IDEA DESCRIPTION FORM

1. Your name, office address, telephone number, e-mail address:

Len Veil , Cupertino, (408) 517-6601, lveil@wavesys.com

2. Title of Invention:

Method and System for Conditional Installation and Execution of Services in A Secure Computing Environment

3. Prior Technology or Methodology (i.e., what has been done before?): Please (1) provide all prior documents (including prior publications, patents, and brochures describing commercial products) relevant to the background of the invention; and (2) describe briefly what was known before in this field.

[1] US Patent #6,092,202 : Method and System for Secure Transactions in a Computer System
[2] Embassy Patent Application : Public Cryptographic Unit and System Therefor, Peter J. Sprague, Greg Kazmierczak, Docket 1162

4. Summary of Invention: Succinctly summarize the invention. What is new? (I.e., How does the present invention differ from and provide an advantage over previous technology?).

This invention provides a mechanism to securely validate the resources required for a security service to execute in a cryptographic tamper resistant co-processor or equivalent environment, and to only install and execute the service if those resources are available. Additionally, this method provides for the necessary mechanisms for matching the security level (strength) of the execution environment to the needs of the security service. These conditional installation methods can additionally be used to incorporate a backend accounting system where business contracts may be enforced based on the ability to install and uninstall the services.

5. Description of the Invention: Attach existing writings, sketches, drawings, or flowcharts describing the invention. If possible, provide a description of the best way of carrying out the invention with reference to flowcharts.

The methods described include (1) the construction of an Embassy applet, where the applet is composed of two parts. The first part of the applet is an applet header, which defines the resources required by that particular security service, where inside the applet header is an Applet Identifier and the associated list of resources required by that particular applet in order to execute. The first part of the applet is cryptographically signed by a Applet Certification Trust Authority. The second part of the applet is the executable, where the executable is pre-pended with the Applet Identifier. The second part of the applet is cryptographically signed by a Applet Certification Trust Authority. The inclusion of the Applet Identifier in both the first and second part provides a cryptographic linkage ensuring that the resources specified are bound to the particular executable. The cryptographic signature on the first and second part of the applet provide integrity checks in order to guarantee that neither the resource information nor the executable have been altered. Optionally, the executable may be cryptographically encrypted in order to protect the executable from being de-compiled and reverse engineered.

The method includes (2) the usage of a cryptographic tamper resistant security co-processor or equivalent,

which is used in a computer data processing environment. This Embassy device is capable of determining the resources that it can provide to an applet, and is capable of interrogating, at run-time, the resources available in its environment.

The method includes (3) the ability to instantiate in the applet header the security certification level of the applet, and also the personalization of the Embassy device for its' particular security certification level that it can offer to a service.

The conditional installation method (4) is the process of determining whether the cryptographic co-processor environment is capable of offering the necessary execution environment to the applet, and only installing the service in the event that the Embassy device has the necessary resources available that the applet requires, and also allowing for the Embassy device to only install the service in the event that the security level of the execution environment meets or exceeds the security level required by the applet. This process is accomplished by (4a) the Embassy device verifying the signatures on the Embassy applet within the tamper resistant boundary, (4b) constructing an environmental match vector of the services required against the services offered, (4c) constructing a security level match vector of the security level offered against the security level required by the applet, and finally (4d) ensuring that the Boolean true conditions exists for both the environmental match vector and the security level match vector. In the event that the true conditions exists, then the requirements of the applet are satisfied and the service may be installed.

The conditional installation method can be further enhanced by providing for (5) a User Interface which allows the owner of the Embassy device to determine the minimum security level of service that the owner will permit to execute in his/her client environment. This provides for the security level vector to be enhanced with the inclusion of the user preferences, where the enhanced security level vector is then modified appropriately by the user security level preferences. Under this scenario, the user enhanced security vector is used in step (4d) above.

Authors note: Should we talk about the installation process? Is that covered in Peters & Greg's work? Is that a separate work?

Once the applet and environment have met the requirements for installation according to (4), and the installation of the service has been completed, the method further provides for (6) the conditional loading of the applet, for execution, into the Embassy device whenever the resources are available. It is possible that other services are presently executing in the Embassy device, and that resources required for execution are currently allocated to those other services. Therefore, for the post-installation loading of the applet, the environmental match vector is modified to be the services required to the current services available, and conditional loading and execution of the applet is based on the Boolean of the current environmental match vector. In the event that a Boolean true condition exists, the applet may load and execute. In the event that a Boolean false condition exists, either the Embassy device or the User must take some corrective action to free up the un-available resources in order to allow the service to execute.

The methods also provide for (7) the ability to provide a back-end server based accounting system that can enforce business contracts based on the successful installation of applets into an Embassy device environment. In the event that the applet is cryptographically encrypted, and that a mechanism exists for the secure communication of the Embassy device with the back-office as provided for in [2], it is possible for the back-end server to determine when an applet has been installed into an Embassy device, and debit the applet providers account for "hosting" charges. This is accomplished by coupling the installation and un-installation processes with an Applet Inventory, whereby the Applet Inventory is the cryptographically signed and time-stamped identification of all applets, by Applet Identifiers, of all current applets in the

Embassy device. This Applet Inventory is thereby used by the backend servers to determine the duration of installation of services into the client environment, and enforce the hosting aspects of the business contracts by applying charges to the requisite Application Provider.

6. Application: What commercial applications do you see for this invention? Are there other possible uses? Are there potential licensees?

Commercial applications include, but are not limited to, the Embassy system . Refer to the Embassy business plan for other possible uses and potential licensees.

7. Date of Invention: When did you first think of the invention? Attach all documentation and/or notebook entries showing the earliest invention date.

January 2000

8. What is the stage of the invention□s development: Has the invention been made, and if so when? Has it been tested, and if so when? Has the invention been placed on sale? If so, when? If not, is there a launch or release date?

The conditional loading methods are presently implemented in the Embassy prototype system since July 2000, and is currently under test. The Embassy system was previously offered for sale, with incremental changes such as this method which are not detailed in the FOR SALE offering. Launch date is set for some time in December 2000.

9. First Disclosure to Others: Has the invention been disclosed to anyone outside the company? If so, to whom, when, where, and under what circumstances?

The current method is shared only under NDA with WAVE SYSTEMS CORP business partners.

10. Publications By the Inventor(s): Has the invention been disclosed in a written publication or is a publication planned? Please attach copies of all publications describing or related to this invention.

These methods are not presently published, and the publication schedule is not set AT THIS TIME. However, the corporate strategy is to publish this work.

11. Agreements with Others: Are there any contracts or agreements with other entities concerning ownership or use of the invention? If so, please attach.

None that I am aware of.

12. Other Information: Please indicate briefly any further information that would help assess the potential for this invention.

None

I/We recognize that under the terms of my/our employment contract, this invention is to be assigned to Wave Systems.

Signed: 1. Leonard Scott Veil — 10/18/00
Inventor — Date

93 Amato Ave
Home Address

Campbell — CA — 95008
City — State — Zip Code

Signed: 2 Erica Elisabeth Tups — 10/18/00
Inventor — Date

1575 Mendenhall Dr. Apt #4
Home Address

San Jose — CA — 95130
City — State — Zip Code

Additional Inventors page - NO

Read and Understood by:

_____ _____
Witness — Date

wave

Wave Systems Corp. Announces EMBASSY Applet Developers Kit

Developer Kit to support Trust@ the Edge applications

Lee, Mass - August 22, 2000 Wave Systems Corp., (Nasdaq: WAVX) today announced the availability of an Applet Developers Kit (ADK) for third party developers. Working in concert with ARM [(LSE: ARM); (Nasdaq: ARMHY)], the industry's leading provider of 16/32-bit embedded RISC microprocessor solutions, Wave Systems has leveraged the ARM Software Development Toolkit (SDT) to create a comprehensive development environment for the EMBASSY security processor. In delivering this environment, Wave Systems has paved the way for third party developers to independently create applications for unique Trust @ the edge applications.

Wave Systems is leading the Trust @ the Edge revolution, a new architectural model which embeds trust and security in user devices and opens the door for a broad range of new services built upon client trust and authentication. A critical element in this architecture is the EMBASSY security processor, an advanced System on a Chip solution with an embedded ARM core. By coupling this powerful core with a variety of additional resources, such as non-volatile storage, unique Ids, flexible I/O, a real-time clock, and cryptography accelerators, Wave Systems has created an open and programmable platform for truly secure processing. This platform will provide secure resources to third party developers for enhancing the security of existing applications, as well as introducing a range of new services that are wholly dependent on a Trusted Client.

"Wave Systems' EMBASSY Trusted Client System is an innovative use of ARM® technology and will enable a new generation of e-commerce, content protection, and secure applications in user devices, " states John Sharp, Development Systems product manager for ARM. " By leveraging our proven software development environment, third party developers will be able to rapidly develop and deliver ARM Powered® applications."

"Our vision is that a wide range of developers will exploit this secure, yet open, platform for delivering enhanced applications to clients," said Len Veil, VP, Engineering of Wave Systems Corp. " This development kit provides a very effective environment for the creation of EMBASSY applets."

Wave System's Applet Developers Kit provides the necessary software and hardware tools to create applications for the EMBASSY system. The kit will include an ARM SDT 2.51 Development Suite, an EMBASSY Applet Development Board with full I/O capabilities, and an ENVOY device powered by EMBASSY. These resources will enable developers to test their core application code, as well as interaction with a variety of system level hardware interfaces, such as USB, ISO-9716, secure input (keyboard, and secure output (LCD). Applications for Beta partners are currently being reviewed and unrestricted distribution of the ADKs will begin in November of this year. Please visit the following URL for additional information on pricing, technical details and the Beta program:

http://www.wave.com/developer

About Wave Systems - Wave Systems' goal is to build a worldwide network of users based on trusted electronic relationships. Trust @ the Edge defines a new architectural model for the Internet, which embeds trust and security in every user device. Wave Systems is developing, deploying and licensing its EMBASSY Trusted Client technology for the mass adoption of this revolutionary model. Integrating industry standard functions from a wide range of partners that enable reliable, secure digital exchange and commerce over the Internet. At its core, Wave Systems is building the services, and enabling 3rd parties to build services that will take advantage of this new, open, Trust @ the Edge model.

About Trust @ the Edge - All current solutions involving, privacy, security and commerce are based on the centralized network portal model. This model takes advantage of PC's for processing, the Internet for connectivity, and the World Wide Web for browser navigation but lacks the one component that will enable the true explosion of digital exchange and commerce, as well as enhance privacy - embedded trust and security in Internet user devices. Trust @ the Edge is revolutionary in that it turns the Internet inside out, moving core security and e-commerce functions out to the edge and places them in the user's Internet device. Trust @ the Edge is the integration of strong security in every user device which provides for the creation of trusted relationships and enables reliable digital exchange and commerce over the Internet.

Safe Harbor for Forward-Looking Statements Except for the statements of historical fact, the information presented herein constitutes forward-looking statements within the meaning of the Private Securities Litigation Reform Act of 1995. Such forward-looking statements involve known and unknown risks, uncertainties and other factors which may cause the actual results, performance or achievements of the company to be materially different from any future results, performance or achievements expressed or implied by such forward-looking statements. Such factors include general economic and business conditions, the ability to fund operations, the loss of market share, changes in consumer buying habits and other factors over which Wave Systems Corp. has little or no control.

Close this window

Stay up-to-date on news relating to Wave. Join our email list by putting your email address in the space below.

join

Prepared for Baker Botts
By: Ed Green, Wave Systems

10/8/2005

Wave®

EMBASSY®
Applet Development Kit (ADK)
Developer's Guide

Part Number: 03-000004

February 2001
Version 1.0

Wave Systems Corporation
480 Pleasant Street
Lee, MA 01238
1-413-243-1600
www.wave.com

EMBASSY Manager Resources Tab

This tab is for informational purposes only. This tab lists all potential resources such as secure input, secure display, biometric sensors, etc. that accompany the EMBASSY Device. Resources supported by the user's particular device are indicated; resources not supported by the user's particular device are grayed out.

Operating system information is also available in this tab.

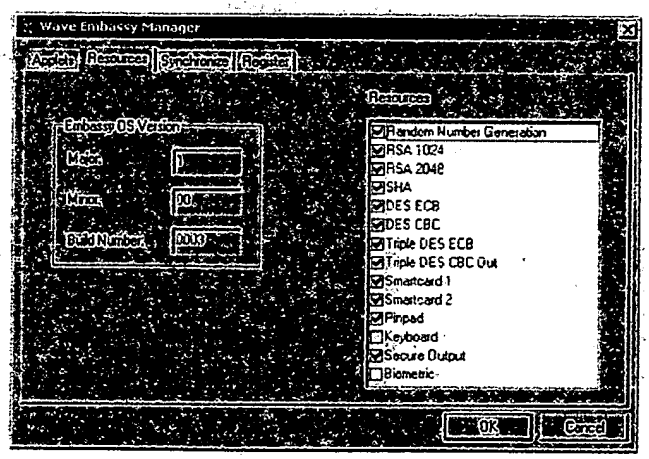

Figure 10-3: EMBASSY Manager Resources Tab

Error: -741

Problem: Too many incorrect password attempts, system locked. Too many failed attempts to enter the password.

Resolution: Reset the password. Reattempt installation.

Error: -759

Problem: Applet requires a resource which is not present on the device.

Resolution: The Applet cannot install onto the device.

Error: -760

Problem: Device OS version too old for Applet to run. The device needs to upgrade to the latest OS.

Resolution: Contact Customer Support for information on how to upgrade to the latest OS.

Applet Publishing Problems

Error: -765

What is an ESAI and where do I get mine?

Problem: When using the ACW, the user is asked for an *.esai file.

Resolution: The ESAI is an Encrypted Secret Applet ID. The user is responsible for creating this secret number (20 bytes in length), and then appending it to the Applet ID. The 24-byte concatenation is then encrypted using the Applet Developer Services (ADS) public key. This key is an RSA public key, which is available in the `\bin\Program Files` directory in the ADK installation.

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0013 (05_7_2001)

The major changes for this release include:
1) Forceunload has been removed from the ADK. If you find it necessary to evict an applet from the Embassy device and cannot do it programmatically, it is recommended that you reset the board.

PRs fixed in this release include:
2262, 2320, 2334, 2355, 2360, 2492, 2527, 2547, 2571, 2581, 2582, 2590, 2592, 2593, 2594, 2595, 2596, 2597, 2609

Known Issues:
No new issues are known at this time.

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0012 (04_6_2001)

The major changes for this release include:
1) We have a new tool for unregistering an Embassy board. The program is called unreg.exe and is a standard part of both the NDEV ADK and XDEV ADK environments. Just run the program, and it will automatically unregister your board. This tool is not available, and will not work in the Test environment.

PRs fixed in this release include:
2195, 2295, 2301, 2302, 2354, 2344, 2345, 2357, 2361, 2392

Known issues:
Installation of Java Runtime Environment fails on Windows 2000 Professional Service Pack 1. The runtime environment is not required for systems running Service Pack 1, so the reported error has no effect on the proper function of the ADK. The PR for this issue is 2243.

The EmbUsb.sys driver may not work on certain systems. Systems with an Intel 82801AA USB Universal Host Controller will experience unrecoverable communication difficulties after loading the OS on the Embassy device. There is no workaround for this problem at this time, except to use the ADK on machines with any other USB controller. If you believe you have this problem, check the USB Host Controller in the Windows Device Manager. If it has the following under the Universal Serial Bus entry:
"Intel 82801AA USB Universal Host Controller",
then you must either install another USB controller and use it, or use a different computer for development. The PR for this issue is 2309.

Upgrading the ADK resets the device resources. To work around this issue, simply run ResConfig.exe and then EDConfigResM.exe again. The PR for this issue is 2386.

Two Embassy Manager icons show up in the Control Panel.A fix for this issue is under investigation. The PR for this issue is 2388.

Time_Difftime_ascii() returns a time that is exactly one day (86400 seconds) off. This is because the leap year calculation was incorrect. The PR for this issue is 2393.

For systems running Windows 98, or Windows 98SE, add "%WINDIR%\system32" to the PATH variable in your autoexec.bat, because some of the files for the installation are not installed to the correct locations. The PR for this issue is 2423.

In some cases, the Embassy Manager Applets page may not reflect what is really installed on the device. This happens when the Embassy Manager is open to the Applets page, and left to sit. During that period, a program may install or uninstall applets unbeknownst to the Embassy Manager. To

assure that the display in the Embassy Manager is up-to-date, click on the "Refresh" button before performing applet maintenence. The PR for this issue is 2439.

When using XPP_Read (also known as Applet_APP_Read), the arguments returned are in reverse order. A quick fix is to modify your applet to note that the two fields have simply been reversed. After you call:

```
XPP_read(uint8 *XPD, uint8*XPID)
```

change your code so that it realizes that:

```
The XPID contains the first 8 bytes of the actual XPD.
The XPD begins with the 9th byte of the  actual XPD
The last 8 bytes of the XPD contain the actual XPID.
```

The PR for this issue is 2441.

The program EDConfigRes.exe is missing from the XDev ADK, so the shortcut in the Start Menu called "Embassy Device Configure Resources" doesn't work. Use EDConfigResM.exe, whose Start Menu shortcut is:

```
"Embassy Device Configure Resources Multiple"
```

instead. The PR for this issue is 2467.

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0011 (04_2_2001)

Please note that there is a separate document detailing changes for the TEST environment. This is necessary because some fixes may occur in one but not both environments. If you are installing the TEST environment, go read test_release_notes.txt!

PRs fixed in this release include:
2188, 2202, 2228, 2260, 2270, 2303

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0010 (03_21_2001)

The major changes for this release include:

1) The debugger privileged applet (debugger.axf) has been restored to the ADK and works properly.

Known Issues:

The ASRand sample applet does not compile properly. In the source files (asrand.c), all references to Applet_XPP_read must be changed to Applet_APP_read, and all references to Applet_XPP_write must be changed to Applet_APP_write, and then the applet will build properly (PR 2260).

Because some components vary between the internal and external versions of the ADK, ADK_version may report some components as not found when the program is run (PR 2317).

PRs fixed in this release include:
1647, 1769, 1901, 2017, 2028, 2057, 2191, 2201, 2206, 2226, 2259

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0009 (03_19_2001)

The major changes for this release include:

1: There are now two versions of the Embassy ADK: an internal version called EmbassyADK_INTERNAL_DEVELOPERS_ONLY.exe, and an external version called EmbassyADK_EXTERNAL_DEVELOPERS.exe. The external version of the ADK requires that the Embassy board has been initialized as an external development board. Once initialized, users can register the board directly and use it normally without

further effort. The internal version of the ADK still requires that the user run fakepers.exe prior to registration if the board was not initialized. To convert a board over to run the external ADK version, perform the following operations:

a) Reset the Embassy board, and plug it into the computer. If the Embassy OS loads automatically, you must go into the registry and delete the following key:

\\HKEY_LOCAL_MACHINE\SOFTWARE\Wave Systems Corp.\Embassy\Embassy OS

If you are unfamiliar with changing registry entries, find someone to assist you with this operation.

b) Run XDevLoader.exe, which can be found in the ADK bin\ProgramFiles directory. Specify XDev.bin for the OS to download, Xdev.sig for the signature file, and internalkeysig.dat for the key file. Once the OS download is complete, you never have to do this again.

c) From this point on, you can run SignEmbassyLoader.exe to download the Embassy OS, using embassy_mug1.bin, embassy_mug1.sig, and oskeysig.dat from the ADK bin\programFiles directory. embassy_mug1.bin is the OS, embassy_mug1.sig is the signature file, and oskesig.dat is the key file.

d) One further requirement is that the device resources must be set. From the ADK bin directory run the ResConfig.exe program with none of the resources selected and a version number of zero. This will create a file named Resources.ed in the bin directory.

Follow this by running the EDConfigResM.exe program. In the privileged applet field type "ProgramFiles\cfgresm.bin" and then press the "Configure ED M" button.
This will clear all entries that the board currently has in its resource table and setup the board for configuration.

Once again run the ResConfig.exe program, but this time select the resource options that match the resources on your board.

Finish by once more running the EDConfigResM.exe program with the same options as before.

Once the resouces on the device have been configured, there should be no need to repeat this procedure.

2: The RSA library functions have changed substantially. Note that you check your calls to this library for conformance with the new interface and you *must* rebuild your applets with the new libraries, because none of your existing applets will work properly with the new library otherwise.

Most functions have some added parameters, or function parameter orders changed to make it more consistent. Two new functions have been added - RSAVerifyHash and RSASignHash. Two functions have been removed - RSA_EncryptWithOAEP and RSA_DecryptWithOAEP. OAEP functionality can now be done using RSAEncrypt and RSADecrypt calls. For more details, refer to the API document.

Modified APIs

1. status Applet_RSA_Decrypt(
uint8 *ciphertext,
uint16 cipherlen,

```
        uint8 *reply,
        uint16 *replylen,
        RSAPrivKey *key,
        const ENCODING  encoding)

2. status Applet_RSA_Encrypt(
        uint8 *plaintext,
        uint16 plainlen,
        uint8 *reply,
        uint16 *replylen,
        RSAPubKey *key,
        const ENCODING  encoding)

3. status Applet_RSA_Verify(
        uint8 *plaintext,
        uint32 plainlen,
        uint8 *signature,
        uint16 signaturelen,
        RSAPubKey *key,
        const ENCODING  encoding,
        BOOL *isValid)

4. status Applet_RSA_VerifyFinal(
        RSASignHandleType* handle,
        uint8 *plaintext,
        uint32 plainlen,
        uint8 *signature,
        uint16 signaturelen,
        RSAPubKey *key,
        const ENCODING  encoding,
        BOOL *isValid)

5. status Applet_RSA_Sign(
        uint8 *plaintext,
        uint32 plainlen,
        uint8 *reply,
        uint16 *replylen,
        RSAPrivKey *key,
        const ENCODING  encoding)

6. status Applet_RSA_SignFinal(
        RSASignHandleType* handle,
        uint8 *plaintext,
        uint32 plainlen,
        uint8 *reply,
        uint16 *replylen,
        RSAPrivKey *key,
        const ENCODING  encoding)
```

New APIs

```
1. status Applet_RSA_SignHash(
        uint8 *hash,
        uint16 hashlen,
        uint8 *reply,
        uint16 *replylen,
        RSAPrivKey *key,
        const ENCODING  encoding)

2. status Applet_RSA_VerifyHash(
        uint8 *hash,
        uint16 hashlen,
        uint8 *reply,
```

```
            uint16 *replylen,
            RSAPubKey *key,
            const ENCODING  encoding,
            BOOL *isValid)
```

3: Do not run fakepers.exe it you are working with an initialized Embassy board, because you will overwrite personalization information and the board will have to be re-initialized.

Known Issues:

There may be issues in synchronizing your device after upgraded to the xdev release. To resolve any synchronization problems, unpower you device and remove and reinsert your battery. Repower the device and try to sync.

PRs fixed in this release include:
1636, 1945, 2005, 2112, 2208, 2248, 2205, 2209, 2054, 2108, 2152, 1860, 2169, 2204

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0008 (NO DATE)

This was an internal version that was never released.

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0007 (NO DATE)

This was an internal version that was never released.

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0006 (NO DATE)

This was an internal version that was never released.

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0005 (NO DATE)

This was an internal version that was never released.

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0004 (NO DATE)

This was an internal version that was never released.

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0003 (NO DATE)

This is the external ADK conversion OS, which can only be run once to convert internal Embassy boards into external Embassy boards. This OS can only be run once on an internal Embassy board, and will report the error "invalid signature" if you run it again on an Embassy board.

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0002 (02_23_2001)

The major changes for this release include:

1: The problem where some environments where registration and installation fails has been resolved.

2: fakemanufacturing.exe, MAL_Stress.exe, nuke_ffs.bat and nuke_ffs.in have been removed from the ADK as unnecessary. To register your Embassy board, perform the following steps:

> Load the OS, if necessary (users of ECase boards must do this using SignEmbassyLoader.exe found in the \bin directory of the ADK). Users with N*Guard development boards can use EmbassyLoader.exe as usual, or autoload the OS by unplugging the USB cable, waiting 5 seconds, and then plugging it back in.

run fakepers.exe, found in the \bin directory of the ADK.

Start the ESE. DO NOT RESET THE Ts!

Register the Embassy board with the Embassy Manager, found in the Control Panel.

a. click on the "Register" tab.
b. Provide a password at least 6 characters long in the "Password" box
c. Probide an answer at least 3 character long in the "Answer" box
d. Click on "Register" button at the bottom.
e. now you should be able to synchronize by click on the "Synchronize" tab then "Synchronize Now"

When changing over from using the ESE to the ENS, it is recommended that you uninstall all applets and then re-register your device with the new service. Each time you transition between ESE and ENS this will be necessary. Note that you will only be able to install published applets when using the ENS. You need to synchronize every time you reload the OS.

3: Some of the sample source code has been removed, since there were no projects or workspaces associated with the code. The following files were removed from the ADK:

adk\sample_applet\src\sample_applet_huge.c
adk\sample_application\src\install_sync.c
adk\sample_application\src\sample_install_sync_test.c

Known Issues:
Icons in the start menu may not appear correctly on systems where a previous version of the ADK was installed. It takes two attempts to change individual icons in the Embassy ADK folder to the proper appearance.

For some environments, it may be necessary to install, uninstall and re-install the ADK for the Java modules to work properly. If you get an class not found error message when starting the ESE, you are probably one of those people.

The ADK may not be usable on original (first release) installations of Windows 98. Some of the updates to Win98 are required for the Embassy Manager to function.

You may experience some difficulties with the debugger.
1. PR 2191 - If you run a debugger version of applet, at the end you may see "error 1071 -
Hot API, error unloading the applet" This may complicate the uninstalls. But you can use ForceUninstall to uninstall the applet
2. PR 2195 - You may not be able to modify some local variables - If a variable is tied to
Register, it will not retain a new value entered by the user through the Local Variables
or User Register windows.
3. PR 2167 - You may experience incorrect branching when set breakpoints. we've only found
in one case. The problem arises as a result of placing a breakpoint on a branch instruction. The user CPSR is not restored properly for the next Go/Step operation
resulting in a possible inappropriate branch. The workaround involves placing breakpoints on the addresses of the two arms of the branch instead of the branch instruction, allowing the correct CPSR value to be used to execute the branch.

If you're using PDS:
PR 2188 - PDS survives applet uninstall if an upgrade is involved. Workaround - install and
uninstall version 0 of the applet.

PRs fixed in this release include:
2182, 2181, 2174, 2110, 2177, 2164

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0001 (02_21_2001)

The major changes for this release include:

1: The issue with registration and installation failing has been resolved.

2: nuke_ffs has been returned to the ADK. The new version is safe to use on all boards (both ECase and N*Guard). You can use the batch file nuke_ffs.bat to reformat the device flash memory if required. After performing this operation, it will be necessary to re-register the device, using the procedure enumerated below:

Load the OS, if necessary (users of ECase boards must do this using SignEmbassyLoader.exe found in the \bin directory of the ADK). Users with N*Guard development boards can use EmbassyLoader.exe as usual, or autoload the OS by unplugging the USB cable, waiting 5 seconds, and then plugging it back in.

run fakemanufacturing.exe, found in the \bin directory of the ADK.

run fakepers.exe, found in the \bin directory of the ADK.

Start the ESE. DO NOT RESET THE Ts!

Register the Embassy board with the Embassy Manager, found in the Control Panel.

a. click on the "Register" tab.
b. Provide a password at least 6 characters long in the "Password" box
c. Probide an answer at least 3 character long in the "Answer" box
d. Click on "Register" button at the bottom.
e. now you should be able to synchronize by click on the "Synchronize" tab then "Synchronize Now"

When changing over from using the ESE to the ENS, it is recommended that you uninstall all applets, or perform a nuke_ffs and then re-register your device. Each time you transition between ESE and ENS this will be necessary. Note that you will only be able to install published applets when using the ENS.

Known Issues:
Icons in the start menu may not appear correctly on systems where a previous version of the ADK was installed. It takes two attempts to change individual icons in the Embassy ADK folder to the proper appearance.

EMBASSY ADK RELEASE NOTES for ADK Release 1.01.0000 (02_09_2001)

The major changes for this release include:

0: DANGER - NEVER USE NUKE_FFS WITH AN ECASE BOARD! IF YOU DO, YOU WILL RENDER IT COMPLETELY AND PERMANENTLY INOPERABLE -- YOU HAVE BEEN WARNED!!!

1. The normal sequence of operations using setup.exe have been superceded. Because the ADK now supports registration, the sequence of events for initializing an Embassy board is only done ONCE (unlike setup.exe, which had to be run every time you reset the Embassy board) and is as follows:

Load the OS, if necessary (users of ECase boards must do this using SignEmbassyLoader.exe found in the \bin directory of the ADK). Users with N*Guard development boards can use EmbassyLoader.exe as usual, or autoload

the OS by unplugging the USB cable, waiting 5 seconds, and then plugging it back in.

run fakemanufacturing.exe, found in the \bin directory of the ADK.

run fakepers.exe, found in the \bin directory of the ADK.

Start the ESE. DO NOT RESET THE Ts!

Register the Embassy board with the Embassy Manager, found in the Control Panel.
Details of this operation are outlined in item 4).

2. Users with ECase boards must still load the Embassy OS manually using SignEmbassyLoader.exe.

3. There are a number of new programs included with the ADK:

fakemanufacturing.exe

fakepers.exe

resetts.exe

It is no longer be necessary to erase flash memory with nuke_ffs, so it has been removed from the ADK.

4. Applet Upgrades, Registration and user authentication are now available. In addition, all the tabs and available functions in the Embassy Manager are operational. Applet installation and upgrade no longer prompt for an applet ID.

Upgrade
The Embassy Mgr's Applet Page now has an Upgrade button. It works similar to installs. It will upgrade from the older version of the applet to the newer version, transfering flash and state data from the older version to the newer version. The older version is uninstalled and the newer version installed.

Registration
a. click on the "Register" tab.
b. Provide a password at least 6 characters long in the "Password" box
c. Probide an answer at least 3 character long in the "Answer" box
d. Click on "Register" button at the bottom.
e. now you should be able to synchronize by click on the "Synchronize" tab then "Synchronize Now"

5. The Embassy OS is now signed. For users of ECase boards with N*Guard daughter cards, it is necessary to run SignEmbassyLoader.exe found in the \bin directory of the ADK. The associated files required to download the signed OS are embassy.sig and oskeysig.dat, which can be found in the \bin directory of the ADK.

6. RSA functions have changed, specifically regarding OAEP padding. The functions are listed here:

```
status RSA_EncryptWithOAEP(
      uint8             *PlainText,
      uint16            ptlength,
      RSAPubKey         *Key,
      uint8             *CypherText,
      uint8             *desKey, //only necessary if method is OAEP_SET,
```

```
otherwise, a null is fine
            const               OAEP_METHOD method);


        status RSA_DecryptWithOAEP(
              uint8             *CypherText,
              uint16            ctlength,
              RSAPrivKey                  *Key,
              uint8             *PlainText,
              uint16            *ptlength,
              uint8             *desKey,
              const             OAEP_METHOD method);
```

The method is an enum, but with the same mode (OAEP_IEEE) defined so that existing code will not need to be changed (I did this because the arm debugger will print out the actual name (instead of the numeric value) of the method if it is an enum, makes life easier).

```
typedef enum oaep_method {
        OAEP_IEEE=0,
        OAEP_P1363=0,
        OAEP_PKCS2=0,
        OAEP_SET
} OAEP_METHOD;
```

The first 3 modes are set to zero in order to alleviate some confusion about which padding modes we use. OAEP_IEEE implements the p1363 standard encoding and mask generation function(MGF1). PKCS #1 v2.0 uses the same padding method and mask generation as does p1363.

6. The Embassy Host API includes a performance enhancement to speed up access to the Embassy board for communication intensive operation. Further performance improvements are pending.

   Two EHAPI APIs have changed from

```
        InstallApplet(CEmbassyApi * pAPI, Applet_ID appletID, const char *
AppletBinaryFullPath);
        UpgradeApplet(CEmbassyApi * pAPI, Applet_ID appletID, const char *
AppletBinaryFullPath);
```

   to

```
        InstallApplet(CEmbassyApi * pAPI, const char * AppletBinaryFullPath);
        UpgradeApplet(CEmbassyApi * pAPI, const char * AppletBinaryFullPath);
```

7. ACW - The "Generate Personalization Data" check box is removed as it was redundant with the "Requires Personalization" checkbox.

   The "Requires Personalization" checkbox is changed to "Requires APP - (Applet Personalization Packet)"

8. Sample source code all builds and runs properly.

9. Error logging is now done separately for some of the ADK components. The resulting data can be found in the \bin\LogFolder directory, and those files should all be included with any bug reports to assist in problem isolation.

10. Icons have changed in the ADK tools.

11. The Embassy ADK Applet Developer's guide (EmbassyADK.pdf) included in this ADK is

the latest interim version, and is currently pre-release.

Known issues:
There is a low-frequency problem that may occur during applet installation where the board may not complete the installation correctly. This issue is under investigation.
If this occurs, reset the board and retry the operation. There is a short-term workaround
 - if you always do a sync in the Embassy Manager less than a minute before installation
of applets, this problem does not occur.

Icons in the start menu may not appear correctly on systems where a previous version of the ADK was installed. It takes two attempts to change individual icons in the Embassy ADK folder to the proper appearance.

EMBASSY ADK RELEASE NOTES for ADK Release 1.000005 (01_19_2001)

The major changes for this release include:
1. There is a new interim version of the ADK documentation called EmbassyADK.pdf.
2. Applets may have the following maximum sizes:
        code space 1 Mb
        constant data space 64 Kb
        allocated memory maximum 64 Kb
        stack space 16 Kb
3. Applet Personalization Data is now supported with both the ESE and the ENS
4. Applet Publication with the Embassy Network Server is available, and also supports XPP.
        Instructions for generating .PAR (Publication requests) files

        1. Load certificate ADSCertChain.der ACSCertChain.der:
                Open acw.exe,
                Click on Action
                Click on Generate ESAI
                Browse to Certificate Chain File path
                Click Import Certificate

                Note: The user does not have to import the certificate, they can use the certificate chain directly if they so wish.

        2. Now Generate ESAI for a specific applet initiated in ENS:
                Now fill Applet number and Applet version
                Click Use imported certificate (Optional)
                Click Generate ESAI

        3. To create .par file (Publication request) :
                From main acw screen,
                Enter Applet name
                Enter Applet Developer name (using 'Wave Systems Corp')
                Enter applet number
                Enter applet version
                (ignore Embassy OS Version for now)
                In the middle, browse to the Object code file path (C:\Program Files\Wave Systems\Embassy ADK\sample_applet\sample_applet1\Release\sample_applet1.axf)
                For Save as - if check Implicit, .par file goes into User directory; otherwise you must
                specify a target directoryFor Encrypted SAI, browse to the patch created in step 2 above.
                C:\Program Files\Wave Systems\Embassy ADK\bin\ProgramFiles\1028-0.esai
                Now check the Required Resources….

RNG, RSA 1024, RSA 2048, SHA1, Des ECB, Des CBC, 3Des ECB, 3Des CBC out, Secure Output (for example)
Click on Create Publication Request, .par file is created in the Save as folder

4. Can save settings by :
Click File,
Click 'save as'
find target folder,
enter name
Click ok, file saved as a .was file

5. ASRand information is now eXecutable Personalization Packets (XPP).
6. The Embassy OS has a fix for boards with RTC battery issues, and supports recovery for
cases where the time has become incorrect or unusable.
7. Changes to the ACW to support applet publication and other new options.
Note that LCD and Secure Output have been consolidated into Secure Display.
The option for applet personalization in required resources is called Requires Personalization.
8. Error codes can be found in the generic_inc directory of the ADK.
9. RSA OAEP padding is unchanged, but will change in the next release. This has no impact on current work.

The following high priority PRs are known to be fixed in this release: 1846, 1852, 1895, 1898, 1917.

Possible problems:
Sometimes the information displayed in the Embassy Manager is not represented correctly.

The statedata_sample applet uses the old ASRand information instead of the new XPP header file,
and does not compile properly without corrections. See the asrand_sample.c file for the proper
implementation and calling conventions.

Version 1.000003 of the ADK contains no major changes from the previous versions.

EMBASSY ADK RELEASE NOTES for ADK Release 1.000002 (12_15_2000)

The major changes in this release include:
1. Support for 1 Mbyte applet
2. Embassy OS autoload (for more details see Kerry Fan of Wave Princeton, kfan@wavesys.com)
3. The library asrand api has changed to xpp, i.e., Applet_ASR_Write has been changed to Applet_XPP_Write, and
Applet_ASR_Read has been changed to Applet_XPP_Read. The documentation for ASRand/XPP lib will be updated later.
You can view the header files in (ADK Folder)\sample_applet\inc\xpp.h and (ADK Folder)\generic_inc\applet.h.
4. Asrand(xpp) is now 128 bytes instead of 24 bytes.
5. For those who have ECase boards: there is another os loader called SignEmbassyLoader.exe that is located
on the bin directory. This can download an OS, as well as the OS Signature file and the OS key data file.
When all three files are download to the ECase, the ECase will then verify whether or not the OS, OS Signature
file and OS key data file matches to certain specifications. If it does verify, then the OS will get loaded
onto the ECase board. If it doesn't get verified, the SigneEmbassyLoader will get a message from the ECase

that the OS, OS Signature file and the OS key data file don't match and the ECase won't load the OS.

The major bugs fixes:
1. EmbassyManager can now uninstall debug applets.
2. Bug fix number 1763: If request for an installed but not loaded applet has been received, the EHAPI should
send the CMD_QUIT command to the currently loaded applet. This functionality is needed to support the systems
where multiple services sharing the same EMBASSY device. Currently the second application receives an error.
3. Bug fix number 1836: Applet_DES_GetKeyWithIV function has been fixed

For those having problems with drivers problems where the computer would revert to Usbe.sys instead of Embusb.sys:
If you have the 08-04-2000 version of the ADK, you need to completely uninstall this version first, then install
the latest version of the ADK. This should remove the driver Usbe.sys from you windows/system32/drivers directory and
the Usbe.inf in windows/inf directory. If this dosen't work, you need to manual check the inf directory and see if there's
a pre-compiled inf called Usbe.pnf. If it's there you must delete it, and restart your computer and try the new drivers
again.

Possible problems:
Some problems may appear and could be a nuisance to a developer. They seem to occur randomly.

1. PR#1871 - Error 805 "the time offset is too large" appears during Install or Uninstall. Workaround: run setup.exe
2. PR#1882 - Unknow IRQ error message displays briefly on LCD. The device continues after the message. No adverse affects
have been noticed yet.
3. PR#1890 - Sometimes synchronizations fail with error 1087. This happens occasionally during Install and Uninstall.
Workround: reload the Embassy OS.

EMBASSY ADK RELEASE NOTES for ADK Release 11_27_2000

The major changes in this release include:
1. The ACW has changed. It only requires the user to enter an Applet number
and Version number to generate an Applet ID.

Limitations:
1. There's a problem in the EmbassyManager when uninstalling applets that are in debug mode. It's
best to use the forceuninstall to uninstall any applets that are in debug mode.

The major bugs fixes:

EmbassyManger:
1. When highlighting the list on the Applet page, the whole row is highlighted rather just
the one column on Applet name.
3. If there's any applets before the EmbassyManger in on, it now shows it after you

execute
    the EmbassyManger.

EMBASSY ADK RELEASE NOTES for ADK Release 10_24_2000

The major changes in this release include:
1. The EmbassyManger is now included. After you installed the ADK,
   the EmbassyManger program is located on the Settings Control Panel.
   You should see an EmbassyManger icon when the Contol Panel is up.

2. The ACW user interface has been changed. The ACW is still in the process
   of adding new functionalities. Please diregard any of new buttons or
   checkboxes on the ACW. The ACW should work fine without them.

3. The standalone forceuninstall has been changed. The format for running
   the forceuninstall should be this:

   forceuninstall [-a NN where NN = Applet Number within appletID] [-v NN where NN
= applet version within appletID]

   For example: if you have an appletID thats 3200, the applet number is 200 and
version number is 0

   To do a forceuninstall before was:

   forceuninstall -a 3200

   it should now be:
   forceuninstall -a 200 -v 0
   or
   forceuninstall -a 200

The major bugs fixed in this release include:

Embassy OS:
The USB suspension has been taken out. This has caused some computers to give
a DEV_IOCTL errors on the host or -551 errors when sending a message.

Limitations:
Same as previous release.

EMBASSY ADK RELEASE NOTES for ADK Release 09_29_2000

The major changes in this release include:

1. Support applet size up to 1M (release or debug) for the ADK release.
   The maximum applet size is 256K in future production release.

The major bugs fixed in this release include:

None

Limitations:

1. Uninstall the previous version of the ADK before this new one.

2. When downloading debugger applet (DebuggerLoader.exe and debugger.axf), observe the yellow LED on the device. If it blinks during the download then you will not be successful when running ADW and making connection with the device.
 My suggestion for now is to restart the whole sequence again.

EMBASSY ADK RELEASE NOTES for ADK Release 09_25_2000

The major changes in this release include:

1. Debugger support. The toolkit now supports the use of ARM's Symbolic Debugger (ADW.EXE). Please refer to the doc "Embassy User Guide" for detailed instructions on how to use the debugger functionality added to this release.
2. The ACW now adds a certificate chain to the bottom of the published applet. This certificate chain is used by the device to verify the signatures of the applet.
3. File pubkeyssig.dat is gone. During installing and syncing a certificate chain is sent from the host to the device to obtain the EDS's public key. This public key is used during the secure communication between the device and the EDS.
4. New USB driver EmbUsb.sys
5. Several new example applets and application to show ADK interaction.
6. include getstatus utility

The major bugs fixed in this release include:

Embassy OS:
 NA
Embassy Host API:
 NA
Embassy USB Driver:
 Drivers from the 9/22/2000 dosen't work properly. This release fixes some of those
 problems
Embassy Toolkit Libs:
 NA

Limitations:

1. IMPORTANT - please uninstall the previous version of the ADK before this new one. In the future the installation script will do this automatically for you.
2. Require Word application from MS Office 2000 suite to read documentation
3. The ARM Debugger must be set to 9600 bps with hearbeats disabled.
4. Support for applets > 64k has not been added. Because adding debug symbols to applets increases their size significantly, developers may have difficulty using the debug facilities provided with this release.
5. When debugging applets in the ARM Symbolic Debugger applet communications with the host are disabled if the developer steps across the messaging library routines. Therefore, to debug applets that communicate with the host pc it is recommended that breakpoints are set after the Messaging library calls and not on the Messaging

library calls. In addition, do not step over Messaging library calls only run to the breakpoints.

6. For some computers, if you switch off the device while the computer is on, and switch the device back on, you might get a driver problem and won't be able to communicate with the device. Try avoiding switching the power off on the device, and if you do, you might be force to restart the computer again to get the driver back online again.

EMBASSY – November 2000 Status

Overview

- **Comdex** – The Embassy team devoted 2 weeks of the month to preparing for the Comdex trade show, including preparation of the systems and integration of applets and associated applet services. Also, PPG Engineering staffed the show floor in order to operate demos and answer technical questions. Approximately 2 weeks of core development time were lost to support this effort.
- **SW development** – The Embassy client team has finished the coding for Release1.0 and is currently in the midst of bug fixes and integration with the Embassy Network Server. The next ADK Release is targeted for December 15th and is planned to have the entire Release1.0 functionality with as many high priority bugs resolved as possible. The system integration is progressing, although approximately 3 weeks behind schedule due to challenging integration issues and Comdex resource reallocation. ENS integration has been completed for applet publishing, applet installation, time synchronization and applets inventory. Embassy Release1.0 should be ready for integration with the EMA (Embassy Metering Application) team on December 15th, without ENS based applet personalization. Limited applet personalization is supported in the 12/15 release of the ESE (ADK).
- **Embassy Architecture** - The procedures for password recovery on the client side were developed and approved. This will allow User's authentication to be added to Release1.1. which is targeted for completion at the end of February 2001. Design impacts for Release1.1 and the associated risk are still pending on the selection of the Embassy Root Certification Authority, and procedures for the Embassy Network Server to transfer secure information into the IBM crypto card.
- **ECase** (N*Guard/MCM) – First run prototypes were provided to us and these are functioning well, albeit the entire MCM fabrication process is still under review due to unacceptable yield. Additional prototypes are expected to be available during the first half of December, and the first 1200 production units during January. Additional MCM manufacturers are under evaluation to minimize risk and to reduce cost.
- **ECase Platforms** – The reference ECase development boards have been checked out and approved, engineering is in the process of handing off production capability to Product Management. The ECase L5 readers PCB are in checkout.
- **ETrust** (N*Guard+) – The netlist has been released to Samsung and we are on track to have the layout signoff on December 15th; release to the fab shortly thereafter; and have the prototypes ready during March 2001 (assuming Samsung commitments to TAT).
- **User Manuals** –ECase and ETrust data sheets were completed and ready for COMDEX along with preliminary ADK and ENS User Manual. Current schedule calls for the completion of the ADK User Manual during December and the ENS Manual during January.

Monthly Milestones

- Demonstrated the ADK at COMDEX
- Completed development to support up to 1 MB applets
- Completed the B0prime library for smart card applications
- Completed ENS-Device integration for time synchronization and applet inventory
- Completed development of the WEB based interface for ADK library testing (Jukebox)
- Completed SW development for the Personalization Station
- Supported National Semiconductor with their Safe Keeper device

- Defined procedures for Applet Publication and Applet Certification
- Completed the User Authentication Specification document
- Supported development of the metering applet, MSCAPI CSP applet, CyberCOMM applet and the TCPA applet
- Received the first ECase prototypes and successfully verified operation
- Released ETrust netlist to Samsung
- Samsung has completed the final floor planning for ETrust
- Started discussion with Wave's operation team to have a personalization station and an ENS up and running in Lee
- Hired and indoctrinated Michael Young as a Program Manager for Embassy
- Hired and indoctrinated Pratap Kesarkar as FAE engineer for East Coast
- Received acceptance for Jiaying Hou, FAE West Coast engineer.

Next Month's Deliverables

- Release the next version of the ADK by December 15th (Release1.0 functionality).
- Release the first ENS version into QA and have this version available for integration with the EMA.
- Have Interpoint ready for ECase risk production.
- Release ETrust into Samsung fab process.
- Complete the ADK User Manual.
- Complete the procedures for security review and associated release procedures.

Issues/Risk

- The timely progression of ENS QA cycle is based on the ability to get additional testing resources during the coming weeks. Cupertino Recruiting has this as their highest priority task.
- ETrust is still at a risk due to Samsung's slow response. Request for additional support remains open.
- In order to have risk production based on the USB interface units on the first mask we need to place the order within a few days of signoff. The company has to take a decision on this by Mid December.
- The ECase prototype yield is not yet satisfactory. Negotiation with Interpoint and Flip Chip continue in order to improve their fabrication process. On-site meeting with Flip Chip scheduled for 12/6 to discuss yield and response issues.
- The Root Key Management decision is required in order to begin implementation of the Embassy Server PKI. Wave's Corporate Business Development team has the responsibility to finalize this issue.
- The Marketing Requirement Specification document for Embassy is not available. In the interim, PPG Engineering continues to develop Embassy according the CyberCOMM, Wave Metering and limited TCPA requirements.
- BXA Approval is required for Release1.0. Wave's Product Marketing team drives this effort.

Embassy Network Server
External Specification

Wave Systems Corp.

Rev. 1.22
5/30/2000

Document Revision History

Date	Version	Description	Author
2/29/00	1.1	Initial release	Mihran Mkrtchian
4/06/00	1.2	Terminology correction, small additions new sections: 5.9 Embassy Network Procedures; Appendix A and D	MM
4/12/00	1.2.0	Corrections	Mihran Mkrtchian
5/01/00	1.21	Tables and Procedures	MM
5/31/00	1.22	Changed 5.5 ENS Management group Added User Account Security and Audit Policy	MM

7.2 Applet Data

	Field Type	Field Name	Max Size (bytes)	Notes
1	Integer	Applet ID	4	See below
1.1	Integer	AppletNumber	4	Max value is 255
2	String	AppletName	255	
3	Object	Organization		
4	Object	AppletDeveloper		UserAccount (who created)
5	Object	AppletCertificationAgent		UserAccount(s)
6	Integer	EMB_Version	4	See below
7	Integer	ResourceReq	4	
8	Integer	Status	4	See below
9	String	ASRand	24	
10	String	ASRandIndx	8	
11	String	AppletCodeKey	24	Encrypted field
12	Integer	UpgradeAuthorizationFlag	4	
13	String	HostName	Max length	
14	String	Description	256	
15	Data	AccountCreationDate	8	
16	Data	LastUpdateDate	8	
17	Object	AppletDeveloperM		PTR to the UserAccount who did latest modification
18	BLOB	AppletBody		

AppletID {

```
    Byte    AppletCertificatioAgent ID;    // The ID of the Certification Agent
    Byte    AppletDeveloperID;             // Applet developer ID
    Byte    AppletNumber;                  // Number 0-15 reserved for privileged applets
    Byte    AppletVersion;                 // Applet version number
```

}

EMB_Version – first two bytes will be used

Status ::= Choice {

Undefined	[0]	-	Undefined stat – no developer or any other agent is assigned; initial status.
InDev	[1]	-	Applet is in development stage
CertPending	[2]	-	Submitted to applet certification agent and pending certification
Certified	[3]	-	Applet is certified by the applet certification agent
Released	[4]	-	Applet finalized and associated with the Applet Code Key
Published	[5]	-	Ready for download and installation. Application Service Provider / Applet Developer may set this status
Revoked	[6]	-	The applet has been revoked from system (the actual body of applet may remain in the system, but ENS will not support Certify, Release and Publish functions.

}

Embassy Applet Model
Software Design Specification

Wave Systems Corp.
1601 So. DeAnza Blvd., Suite 200
Cupertino, CA 95014
408-517-6600

Deleted: EMBASSY: Applet Model Software Design Specification¶

CONFIDENTIAL

SCOPE: This document defines the applet model of the EMBASSY system. It is intended to communicate the life cycle of Embassy applets, the format of Embassy applets, the execution model of Embassy applets, and the usage of Embassy applets in the Host Application context.

Wave®

Document Revision History

Date	Version	Description	Author
10/13/2000	**2.7**	Tn and TnE changed from Random Numbers to DES keys. Applet Certification Applet Code Signature – removed Page#0. Removed some of the Future Releases feature. Cannot have a synchronization duration greater than 30 days. Changed the Applet Distribution Format (changed the ASN.1 format). Changed the Resource Requirements in the applet header. Changed the length of the Applet Personalization Data.	Erica Tups

An EH application communicates with an applet through commands and command handlers. A command is sent from the EH's application to the ED's OS via the Embassy Host API (EHAPI). The ED's OS passes the command to the applet. Upon receipt of the command, the applet executes a command-handler. A command-handler is the code in the applet which processes the command. The applet passes a response to the ED OS which sends the response to the EHAPI which passes the response to the EH's application. The EH's application is the master and the applet is the slave. In other words, the EH's application initiates communication with the applet and then the applet responds.

Command handlers must always return some type of response to the EH's application.

All system calls must be accessed through system call wrappers. Applet code and static libraries cannot make system calls directly but must use the system call wrappers' API.

If an applet does not (e.g. forgets to) free a resource, the ED OS eventually frees the resource based on time limits.

STATE DATA

An applet may save up to 8K of state data on the EH. State data is data the applet does not want to lose in between loads of the applet. API calls are provided to the Application Provider to:

- retrieve the starting address of the state data
- mark the state data as dirty
- check if the state data is dirty.

When the state data is marked as dirty by the applet, the following steps happens:
The next time a response is sent back from the applet to the EH's application, the ED OS checks if the state data is marked as dirty. If dirty, the response is sent back to the EHAPI with a flag set. When the EHAPI get the response and sees the flag is set, the EHAPI sends a command to the ED's OS to encrypt and send the state data to the EHAPI. The ED OS encrypts the state data with the applet's swap key, saves a hash of the data, and sends the encrypted state data to the EHAPI. The EHAPI saves the state data on the EH and sends an acknowledgement to the ED. The ED marks the state data as clean and returns a successful response to the EHAPI. The EHAPI delivers the applet's response to the EH's application.

APPLET HEADER

When creating an applet, the Application Provider utilizes the Applet Creation Wizard (ACW) to create an applet header. The applet header is never encrypted. The applet header is signed by the ACA. Within the applet header is:

EMBASSY Device OS Version = Which EMBASSY Device OS version is the minumum version this applet can execute on.

Applet Name = String describing name of the applet

Applet Developer = String describing applet developer

Applet ID = Unique ID of applet. The Applet ID consists of:

Bits (0 is leftmost)	Description	Range
0 – 2	Reserved	Must be 0
3 – 25	Applet Number	0 – 8388607
26 – 31	Applet Version	0 – 63

The Applet Number is assigned by the ADS during the initial publication of a particular applet (i.e. an applet with no prior version). The developer requests a new applet on the ADS which will generate a new applet number.

The Applet Version is assigned and managed by the AP. It has the following restrictions:

The initial version of an applet must have a version of 0.
The version must monotonically increase. In other words, the upgrade version number must be larger than the previous version number.

Applet Resource Requirements = The resources required by the applet to execute are indicated. The following resources are available in an Embassy Device:

Personalization Required - This bit indicates that the applet requires personalization. That is, during the installation of the applet an Applet Personalization Pack is sent to the applet from the EDS.

Biometric Sensor
Secure Output
Keyboard
PINPad
SmartCard 2
SmartCard 1
3DES CBC Outer
3DES CBC Inner
3DES EBC
DES CBC
DES ECB
SHA1
RSA 2048 Bit Keys
RSA 1024 Bit Keys
Random Number Generator

8.2 Applet Certification

After successfully certifying an applet, the ACA signs the applet header with its private key. The ACA also signs the applet code with its private key. The ACA actually signs the applet code prepended with the AppletID . The signature is over: SO[AppletID|Applet-Code]ACAPrivK.

If the ED is not yet Registered, the installation cannot take place and is aborted.

If an applet happens to be loaded on the ED, the EHAPI unloads the applet via the QUIT command.

If a synchronization with the EDS is required, the Synchronization applet is first loaded and executed. A synchronization with the EDS could be necessary because an earlier install, upgrade or uninstall transaction was started and not completed. A synchronization with the EDS could be necessary because more than 30 days have passed since the last synchronization with the EDS. A synchronization with the EDS could be necessary because the battery was removed from the clock and the clock must be reset. Also the last synchronization attempts with the EDS could have returned a time offset which is too large for the ED to accept. If the synchronization is unsuccessful, the installation cannot take place and is aborted.

The Installation applet is loaded and executes.

The ED checks that the maximum number of allowed applets to be installed (determined by the device's capabilities) will not be exceeded if the current applet is installed. If the maximum number of applets is already installed, the installation is aborted.

The ACA 's certificate chain is downloaded to the Installation applet and verified using the hash of the Embassy Root Public Key. The ACA's Public Key is retrieved.

The applet header is downloaded. It's signature is verified using the ACA's Public Key.

The applet ID is retrieved from the applet header.

The Resource Requirements are retrieved from the applet header. Verification is made to ensure the resources are available.

The minimum OS version required for the applet to execute is retrieved from the applet header. Verification is made to ensure the device's OS is a version equal to or greater than this requirement.

The applet ID is sent to the EDS in a secure communications message.

The EDS sends back to the EDS the applet key and the APP if one exists.

The applet key is used to decrypt the downloaded applet code.

The ACA's Public Key is used to verify the signature of the decrypted applet code. When the device verifies the signature of the applet, the appletID and Page#0 are prepended to the applet. The device verifies the signature of [appletID,0,applet-code].

The ED creates a local swap key to re-encrypt the applet code. The ED also creates a hash of the applet code. The re-encrypted applet code is sent back to the host and stored on the host.

If present, the APP is saved on the ED. API calls are provided for the applet to manipulate the APP.

The swap key, hash of the applet code, applet ID, and resource requirements are saved on the ED.

When the Installation applet completes, the EHAPI loads the Synchronization applet. The ED synchronizes with the EDS making the EDS aware an applet was installed.

10.3 Applet Load

The following steps are taken during the loading of an applet:

12 Future Releases

The following features are not part of EMBASSY version 1. The following features are scheduled for implementation in a future EMBASSY release.

12.1 Export Control

In a future release, at installation time of an applet, cryptographic functions may be disabled/enabled for that applet based on necessary information delivered by the ES. This will allow cryptographic strength to be determined on a per applet basis.

12.2 Synchronization Duration

If the ES wishes to change the number of days in between synchronizations, the synchronization message from the ES to the device indicates the new parameter. The device stores and acts upon the new duration. The duration cannot be greater than 30 days.

12.3 Synchronization/Device OS Version

When the ED synchronizes with the EDS, the ED sends the EDS the current OS version the ED is executing. This allows the EDS to determine if an OS upgrade is a) not necessary, b.) recommended or c.) required.

12.4 Resource Requirements

In a future release, multiple applets will be loaded and executing at one time.

An applet will not be loaded if a required resource is reserved by another applet. For example, if applet A is executing and reserves a resource, then applet B attempts to be loaded, if applet B needs to reserve the same resource, applet B will not be loaded and an error will be returned to the EH application.

12.5 System Access Control List

Wave Systems Announces EMBASSY® Trust System

Comprehensive EMBASSY System Is the Industry's Most Advanced for Embedding Programmable Multi-Party Trust System in User Devices

Lee, MA - April 10, 2001: - Wave Systems Corp. (Nasdaq: WAVX - www.wave.com) today announced the availability of the EMBASSY Trust System, the industry's first comprehensive trust infrastructure, tools and device components required to create, deploy, and manage Trusted Client hardware devices which are open, shared and programmable.

While single function security devices are prevalent today, the breakthrough architecture achieved by the EMBASSY Trust System provides the capability for multiple entities, such as service providers, content owners, and individuals, to share a single device while trusting that their individual interests have each been strongly protected from both local and network sources of attack.

The EMBASSY Trust System enables the establishment of a network of hardware Trusted Clients - embedded in personal computers, peripherals such as smart card readers and keyboards, set top boxes and Personal Digital Assistants, as well as the development, deployment and installation of cryptographically protected applets - software applications - representing independent service providers in the platform.

The comprehensive EMBASSY Trust System has been designed to provide the security strength of hardware, but the flexibility of software, solving both a full range of today's security problems as well as providing a distributed, open, and fully programmable platform to exploit tomorrow's opportunities as privacy and security technology evolves and advances. This flexibility allows the EMBASSY to easily fulfill evolving security requirements such as those specified by the Trusted Computing Platform Alliance(TCPA) for adding security into the PC platform, while also addressing a broad range of advanced applications which cannot be solved by the TCPA specifications, such as content protection, user managed privacy, strong authentication of user identities, and distributed e-commerce.

"The EMBASSY Trust System is a unique, comprehensive offering for the world of e-commerce where companies and consumers will have the ability to securely and privately transact business, access sensitive medical and education records and be innovatively entertained on the Internet," said Steven Sprague, president and CEO, Wave Systems.

The EMBASSY Trust System (ETS) consists of the following major elements:

- EMBASSY client hardware chips
- EMBASSY client Trusted Operating System (TOS) software
- Trust Assurance Network (TAN), a trust delivery infrastructure of servers
- Software and Hardware Developer's Kits for EMBASSY devices and applets

The TAN provides a complete set of server functions based on an independently managed root of trust for creating, deploying, and operating a trusted network of devices and secure applications to run inside these Trusted Client devices. In March, EDS was announced as the Root Key holder for the EMBASSY Trust Systems, which are expected to be licensed and deployed to a number of enterprises and trust service providers this year.

This innovative Trust @ the Edge architecture which specifies embedded trust in every user

device and peripheral for the Internet enables secure applications, services, and electronic commerce, while simultaneously providing a solution for user-managed privacy.

The complete EMBASSY Trust System announced today also includes the EMBASSY Applet Development Kit (ADK) for third party development of applications using the resources of the EMBASSY System, and the EMBASSY Hardware Development Kit (HDK) for hardware partners wishing to integrate EMBASSY technology into their client computing environments.

The EMBASSY Trusted Client
The EMBASSY Trusted Client is the heart of the EMBASSY Trust System and the basis of the Trusted @ the Edge Internet architecture. The EMBASSY client device provides an ARM processor, cryptographic accelerators, secure I/O capabilities, a trusted real time clock, and secure execution of applets providing a "system within a system" to protect sensitive operations and data from attacks through the host's (PC/PDA/set top box) operating system and hardware. The EMBASSY chip, operating under control of its trusted operating system, provides the secure execution environment for the sensitive portions of an application (applet) to run within. These applets can easily be upgraded as new versions of an application are distributed or as security, content protection, or other industry standards evolve

EMBASSY Trusted Operating System
EMBASSY devices include the EMBASSY Trusted Operating System (TOS), which enables the authentication, secure loading and unloading, and protected execution of the signed and encrypted applets. The Trusted OS itself can be field upgraded for adding new features, algorithms and renewable security as required. For many host applications access to secure hardware resources such as a real time clock and non-volatile storage may be the only requirements. Applications can also be developed using the rich set of services of the Trusted Operating System, or they can leverage the capabilities of industry standard security APIs which have been ported to the Embassy client environment.

EMBASSY Trust Assurance Network
Wave's EMBASSY Trust Assurance Network (TAN) will provide a complete trust infrastructure required to support and operate a network of Trusted client devices and secure applets. Features of the TAN include EMBASSY device initialization, registration and synchronization, applet publishing, certification, installation, revocation, upgrades, inventory, and personalization. Wave Systems will operate a TAN and will also license components of the TAN to third parties for either publicly available TAN services or for enterprises who would prefer to host their own EMBASSY trust infrastructure.

TAN Servers
The TAN license covers the Application Developer Server for application providers to manage their applets for the EMBASSY system, The Applet Certification Server for third party Applet Certification Services, and the EMBASSY Device Server, the key enabler and authenticator of EMBASSY devices for the EMBASSY system. Integral components of TAN also include the Authorization Server, a secure environment for the creation of unique IDs for newly manufactured EMBASSY devices, and the Initialization Station to initialize EMBASSY devices during the manufacturing process. As part of the EMBASSY Trust System Wave also announced a complete set of tools for the creation of Trusted applications, and Trusted Client devices.

EMBASSY Applet Development Kit (ADK) and Hardware Development Kit (HDK)
Wave's EMBASSY Applet Development Kit (ADK) provides all the tools and components necessary for third party developers to code, test, and develop applications utilizing the secure resources of the EMBASSY System. ADK components include an EMBASSY Development Board, Device Libraries and the ARM Software Development Toolkit v2.51.

Wave offers an EMBASSY Hardware Development Kit (HDK) for hardware partners who want to design and manufacture their own EMBASSY devices incorporating EMBASSY security chip technology. HDK components include EMBASSY Reference Design, Evaluation Prototype, Client Software, Manufacturing Test Support, and Minor OS upgrades.

About Wave Systems Wave Systems' goal is to enable a worldwide network of users based on trusted electronic relationships. Trust @ the Edge defines a new architectural model for the Internet, which embeds trust and security in every user device. Wave Systems is developing, deploying and licensing its EMBASSY Trusted Client technology for the mass adoption of this revolutionary model. Wave is integrating industry standard functions from a wide range of partners that enable reliable, secure digital exchange and commerce. Wave Systems and third parties are building the services that will take advantage of this open model.

Safe Harbor for Forward-Looking Statements Except for the statements of historical fact, the information presented herein constitutes forward-looking statements within the meaning of the Private Securities Litigation Reform Act of 1995. Such forward-looking statements involve known and unknown risks, uncertainties and other factors which may cause the actual results, performance or achievements of the company to be materially different from any future results, performance or achievements expressed or implied by such forward-looking statements. Such factors include general economic and business conditions, the ability to fund operations, the ability to forge partnerships required for deployment, changes in consumer and corporate buying habits, chip development and production, the rapid pace of change in the technology industry and other factors over which Wave Systems Corp. has little or no control. Wave Systems assumes no obligation to publicly update or revise any forward-looking statements.

Wave Corporate Contact:
Wave Systems Corp.
John Callahan
413-243-7029
jcallahan@wavesys.com

Wave Investor Relations:
Jaffoni & Collins
David Collins, Richard Land
212-835-8500
wavx@jcir.com

Close this window

Stay up-to-date on news relating to Wave. Join our email list by putting your email address in the space below.

join